MEMORY-AUGMENTED VARIATIONAL ADAPTATION FOR ONLINE FEW-SHOT SEGMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

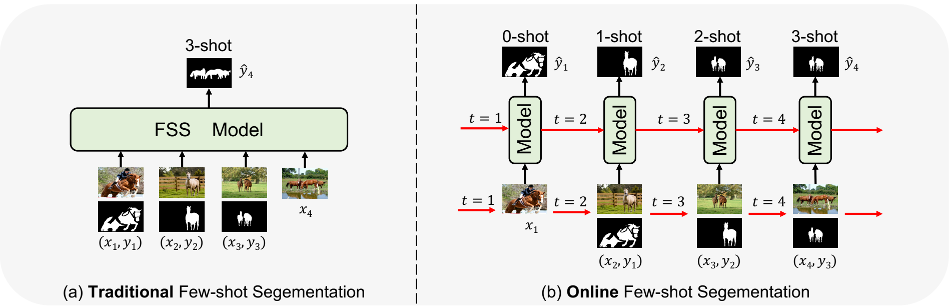
We investigate *online few-show segmentation*, which learns to make dense predictions for novel classes while observing samples sequentially. The main challenge in such an online scenario is the sample diversity in the sequence, resulting in models that do not generalize well to future samples. To this end, we propose a memory-augmented variational adaptation mechanism, which learns to adapt the model to every new sample that arrives sequentially. Specifically, we first introduce a prototype memory, which retains category knowledge from previous samples to facilitate the model adaptation to future samples. The adaptation to each new sample is then formulated as a variational Bayesian inference problem, which strives to generate sample-specific model parameters by conditioning the sample and the prototype memory. Furthermore, we propose memory-augmented segmentation to learn sample-specific feature representation for better adaptation to the segmentation of each sample. With extensive experiments, we show that a simple extension of existing few-shot segmentation methods tends to converge to over-smoothed, averaged masks of lesser performance. By contrast, the proposed method achieves considerably better online few-shot segmentation performance.

1 INTRODUCTION

Few-shot semantic segmentation (FSS) (Shaban et al., 2017; Dong & Xing, 2018; Zhang et al., 2019b; Wang et al., 2019; Yang et al., 2020; Liu et al., 2020; Tian et al., 2020; Min et al., 2021; Zhang et al., 2021a; Liu et al., 2022) learns to segment objects from previously unseen classes, by providing models with a small set of annotated examples, i.e., the *support set*. This setup rests on the assumption that these annotated support samples are revealed to the model simultaneously, e.g., three labelled samples are available under the 3-shot setting in Figure 1 (a). Yet, expecting multiple annotated samples simultaneously in the dynamic world is an unrealistic requirement. For instance, self-learning robotic agents interact with the world in an online manner. And, in a medical setting (Luo et al., 2021) often the expert annotators must correct the models sequentially.

In this work, we investigate the online few-shot segmentation task, which aims to make a pixel-wise prediction for novel classes with samples arriving sequentially. More specifically, the model can only access one sample at a time step in this setting, while the corresponding ground truth comes at the next time step. We clarify the task with an example in Figure 1 (b), where the model is asked to segment the horse in the sequence. At the second time step, the sample x_2 and the ground-truth y_1 are revealed to the model. The model then is updated given (x_1, y_1) and required to segment the new sample x_2. More generally, at time step t the model is updated after seeing $\{(x_i, y_i)\}_{i=1,\dots,t}$ and then makes a prediction for x_{t+1}. In such a way, the model evaluation and updating proceed alternately, and the model learns to segment samples from novel classes in an online manner. Ideally, with increasing samples, an optimal online few-shot segmentation model should become better and better and exhibit more minor performance fluctuations. However, sample diversity in the sequence is the main challenge in such an online scenario (Finn et al., 2019; Babu et al., 2021), resulting in models that do not generalize well to future samples. Therefore, achieving effective model adaptation to each sample in the sequence is essential for online few-shot segmentation.

In this paper, we focus on the online few-shot segmentation (Babu et al., 2021) and make three important contributions. First, we construct a prototype memory based on the idea of "online" class prototype generation (Ren et al., 2020; Finn et al., 2019) that retains category knowledge from



(a) **Traditional** Few-shot Segementation

(b) **Online** Few-shot Segementation

Figure 1: **Comparison between (a) traditional few-shot segmentation and (b) online few-shot segmentation.** under 1-way 3-shot setting. Traditional few-shot segmentation aims to make prediction for one image based on three annotated samples, while online few-shot segmentation copes with the segmentation of all samples sequentially.

previous samples and serves as dynamic support set for the segmentation of future samples. New class prototypes are approximated using groups of same-class exemplar embeddings in the current sequence and stored in external memory. Second, we propose variational test-time adaptation for the online few-shot segmentation, where we formulate the adaptation as the variational inference of a latent classifier variable. For the test-time adaptation, we incorporate the category knowledge from the prototype memory and sample-specific context from the current sample to generate a sample-specific classifier. Furthermore, the probabilistic classifier obtained are more informative and therefore better represent categories of objects compared to deterministic vector. As the third contribution, we propose memory-augmented segmentation to learn sample-specific feature representations better adapted to each sample segmentation. By doing so, the model is endowed with the ability to provide sample-specific segmentation for each sample in the sequence and copes with sample diversity well. Once trained on seen classes, our model could adapt to each sample from novel classes in the sequence with just a feed-forward computation at test time.

We demonstrate the effectiveness of the memory-augmented variational adaptation by conducting experiments on both natural image and medical image datasets. We show that a simple extension of existing few-shot segmentation methods tends to converge to over-smoothed, averaged masks of lesser performance. Our model exhibits promising performance under the online few-shot segmentation setting. Extensive ablation studies demonstrate the contributions of different components in our model.

2 Methodology

2.1 Problem Statement

Few-shot semantic segmentation: Traditional few-shot semantic segmentation (FSS) follows the meta-learning paradigm, where a task (or episode) is composed of a support set S and a query set Q. Here, we consider the 1-way k-shot setting. Conditioned on the support set with k annotated support samples, the few-shot learner $f(\cdot)$ is expected to make pixel-wise prediction for the query sample x^q: $\hat{y}^q = f(x^q; (x_1^s, y_1^s), \dots, (x_k^s, y_k^s))$, where x is input image, y is corresponding binary mask. However, this setup is built on the assumption that annotated support examples are revealed to the model simultaneously, which is usually not practical in our dynamic world.

Online few-shot semantic segmentation: Online few-shot semantic segmentation (OFSS) aims to make pixel-wise prediction on a stream of samples from novel classes. One task consists of T samples from the same novel class. In this setup, samples are revealed to the model sequentially, while corresponding masks are given afterwards. The few-shot learner in OFSS aims to tackle the sequential decision problem: $\hat{y}_t = f(x_t; (x_1, \text{null}), (x_2, y_1), \dots, (x_{t-1}, y_{t-2}))$, where *null* indicates no mask for the first sample, and the model makes a random prediction for x_1. If not specified, we set $t > 1$ for illustration in the following text. By feeding sequential samples and subsequent labels to the model, we evaluate the model online while updating model parameters dynamically.

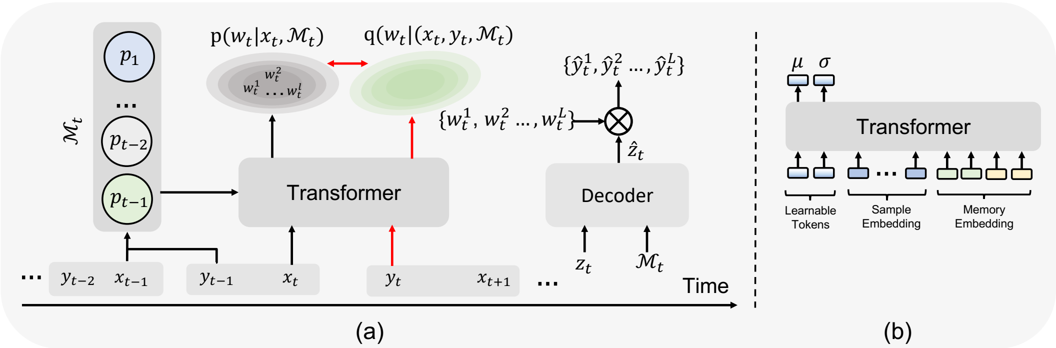


Figure 2: **a) Architecture of memory-augmented variational adaptation for online few-shot segmentation**. At i-th time step, the sample x_t and the previous sample's label y_{t-1} are revealed to the model, which first stores sample prototype p_{t-1} into the prototype memory $\mathcal{M}_t$. Then, the model generates distributions of classifiers via transformer. Lastly, sample feature z_t is enriched with prototype memory as $\hat{z}_t$, which further multiplies sample classifiers $\{w_t^1, w_t^2, ..., w_t^L\}$ from prior distribution $p(w_t|x_t, \mathcal{M}_t)$ to get predicted masks $\{y_t^1, y_t^2, ..., y_t^L\}$. Red arrows are only valid in training. **b) Details of distribution generation with transformer** Learnable tokens interact with sample and memory embedding in transformer to generate distribution parameterized by μ and σ.

2.2 MODEL

We propose a memory-augmented variational adaptation mechanism for online few-shot segmentation. The proposed method achieves online few-shot segmentation via three key components: 1) A **prototype memory** that retains category knowledge from previous samples to facilitate model adaptation to future samples. 2) **Variational test-time adaptation** which formulates new samples adaptation as a variational Bayesian inference problem. 3) **Memory-augmented segmentation** that learns sample-specific feature representation for each sample. We show our model in Figure 2.

Prototype memory Online few-shot segmentation involves segmenting samples from the same classes sequentially. Therefore, effectively leveraging acquired category knowledge from previous samples to boost the segmentation of future samples is crucial. Here, we construct a prototype memory to achieve this goal. Considering memory efficiency, we choose to store sample prototypes rather than original samples in the memory. Specifically, a sample x with ground-truth y can be represented by a sample prototype $p \in \mathbb{R}^{N \times C}$, which is composed of N prototypes with C dimension, respectively. Given a deep neural network $\Phi : \mathcal{X} \to \mathcal{Z}$, which maps from input space to feature space, the sample prototype p can be modeled as clustering centers of foreground features:

$$p = \mathcal{A}(\Phi(x) \cdot y) = \mathcal{A}(z \cdot y), \tag{1}$$

where $\mathcal{A}$ is a clustering function (e.g., K-means), $z \in \mathbb{R}^{C \times H \times W}$ is the sample feature, H and W are the height and the weight, respectively. We adopt element-wise multiplication between z and y to generate foreground features. At the time step t, the ground-truth y_{t-1} of sample x_{t-1} is revealed to the model so that we can store the sample prototype p_{t-1} of sample x_{t-1} into the memory. Similarly, we can store prototypes of all previous samples in the memory $\mathcal{M}_t = \{p_i\}_{i=1}^{t-1}$ sequentially. The prototype memory, which stores prototypes of previous samples sequentially, works as dynamic support set for the segmentation of future samples. Aggregating category knowledge (e.g., different appearance of objects) from previous samples, the prototype memory benefits the model adaption to future samples.

Variational test-time adaptation Sample diversity in the sequence, e.g., large object appearances, is the main challenge for online few-shot segmentation. An optimal online few-shot segmentation model should have the capacity to adapt flexibly to the segmentation task of each sample. To this end, we propose to generate a sample-specific model for each new sample by formulating the adaptation to a new sample as a variational Bayesian inference problem. Our online few-shot segmentation model is composed of a frozen backbone, a decoder network, and a classifier w. At time step t, rather than generating all model parameters, we generate sample-specific classifier weights w_t for the sample x_t, maximizing the conditional predictive log-likelihood $\log p(y_t|x_t, \mathcal{M}_t)$. By incorporating the sample-specific classifier w_t into the prediction distribution, we have

$$\log p(y_t|x_t, \mathcal{M}_t) = \log \int p(y_t|x_t, w_t)p(w_t|\mathcal{M}_t)dw_t, \tag{2}$$

where $p(w_t|\mathcal{M}_t)$ denotes the conditional prior distribution over w_t. By depending on the prototype memory $\mathcal{M}_t$, we infer the classifier w_t aggregating category knowledge from previous samples.

Although the prototype memory provides some category information, the model still knows little about the current sample, especially when previous and current samples exhibit large appearance variations. We cannot guarantee that the prior distribution $p(w_t|\mathcal{M}_t)$ could quickly adapt to the segmentation of the current sample x_t. To cope with this problem, we expect our model to learn to acquire sample-specific information from the current sample x_t. Therefore, we further propose to adapt the model by taking the information from the current sample into account. Instead of using $p(w_t|\mathcal{M}_t)$ as prior distribution, we incorporate the current sample x_t into the prior distribution, so Eq. (2) can be further formulated as

$$\log p(y_t|x_t, \mathcal{M}_t) = \log \int p(y_t|x_t, w_t)p(w_t|x_t, \mathcal{M}_t)dw_t. \tag{3}$$

Conditioned on the current sample x_t and the external memory $\mathcal{M}_t$, the prior distribution $p(w_t|x_t, \mathcal{M}_t)$ aggregates category knowledge from external memory and sample-specific knowledge from the current sample. To guarantee that the prior distribution $p(w_t|x_t, \mathcal{M}_t)$ could generate sample-specific classifier parameters, we design a variational posterior distribution $q(w_t|x_t, y_t, \mathcal{M}_t)$. By incorporating $q(w_t|x_t, y_t, \mathcal{M}_t)$ into Eq. (3), we derive a lower bound of the conditional predictive log-likelihood:

$$\begin{aligned}\log p(y_t|x_t, \mathcal{M}_t) &= \log \int p(y_t|x_t, w_t)p(w_t|x_t, \mathcal{M}_t)dw_t \\ &\geq \mathbb{E}_{q(w_t|x_t,y_t,\mathcal{M}_t)}[\log p(y_t|x_t, w_t)] - \mathbb{D}_{KL}[q(w_t|x_t, y_t, \mathcal{M}_t)||p(w_t|x_t, \mathcal{M}_t)].\end{aligned} \tag{4}$$

This formulation establishes a variational inference of the prior distribution. In such a way, we guarantee the inferred classifiers to be discriminative and adaptive to the segmentation of the current sample. Besides, the KL divergence term in Eq. (4) further works as a regularizer, pushing the prior distribution to adapt better to the current sample. In practice, we generate the prior and the posterior distribution via a vanilla transformer (Vaswani et al., 2017) to allow the flexibility of variable input sizes of conditions. The derivation of Eq. (4) is provided in Appendix A.

Memory-augmented segmentation Semantic segmentation requires much contextual information in the feature representation to make a precise pixel-wise prediction. In our case, we strive to learn better representations with the prototype memory and the current sample. At time step t, we have a sample x_t, initial feature representation z_t, and the prototype memory $\mathcal{M}_t$. Specifically, we incorporate the object context from the prototype memory by introducing a category prototype p_t^c:

$$p_t^c = \frac{1}{t-1}\sum_{i=1}^{t-1} m_i p_i, \tag{5}$$

where m_i is the weight for the prototype p_i. Under the online setting, we can get the prediction mask $\hat{y}_{t-1}$ at time step $t-1$, and the ground-truth mask y_{t-1} of the sample x_{t-1} is revealed to the model at time t. To this end, we can obtain the weight m_i by calculating the cross entropy between the ground-truth mask y_i and prediction mask $\hat{y}_i$:

$$m_i = 1 - \frac{-y_i \log \hat{y}_i}{\sum\limits_{j=1}^{t-1} -y_k \log \hat{y}_j}. \tag{6}$$

With larger m_i, the model is more confident about the current segmentation, then p_i from memory could provide more abundant category knowledge. The category prototype p_{t-1}^c is updated at each time step and expected to obtain robust and generalizable class representation with the time step increases. Then we can obtain the updated feature representation $\hat{z}_t$ of sample x_t:

$$\hat{z}_t = \Psi([z_t, \tilde{p}_t^c, y_t^*]), \tag{7}$$

where Ψ is the decoder network implemented with multiple convolutional layers, $\tilde{p}_t^c$ is the expanding variant of p_t^c with same spatial dimension as z_t. y_t^* is the prior mask of sample x_t modelled by pixel-wise cosine similarity between class prototype p_t^c and initial feature representation z_t. $[\cdot]$ indicates the concatenation operation in the channel dimension. The initial feature representation z_t and prior mask y_t^* provide sample-specific context from current sample x_t, while the category prototype $\tilde{p}_t^c$ contains category knowledge from previous samples. In such a way, we learn sample-specific feature representation for better adaptation to the segmentation of each sample. With sample-specific representation $\hat{z}_t$, we can directly obtain the predicted mask of sample x_t: $\hat{y}_t = \frac{1}{L}\sum_{l=1}^{L}\hat{z}_t w_t^l$, where $w_t^l \sim p(w_t|z_t, \mathcal{M}_t)$. L is number of Monte Carlo samples.

Objective The loss function is computed after all the segmentation tasks in the sequence are completed. We freeze parameters in the backbone network to avoid the model over-fitting over training classes, while the remaining network parameters are optimized end-to-end. By incorporating feature representation z_t and $\hat{z}_t$ into the evidence lower bound in Eq. (4), the final objective function is formulated as:

$$\mathcal{L} = \frac{1}{T}\sum_{t=1}^{T}\Big[\frac{1}{L}\sum_{l=1}^{L}[-\log p(y_t|\hat{z}_t, w_t^l)] + \mathbb{D}_{KL}(q(w_t|z_t, y_t, \mathcal{M}_t))||p(w_t|z_t, \mathcal{M}_t))\Big], \tag{8}$$

where T is the length of sequences. To enable back propagation, we adopt the reparameterization trick (Kingma & Welling, 2013) for sampling the classifier w_t. In practice, the first log-likelihood term is implemented as a cross entropy loss between predictions and ground-truth. The conditional probabilistic distributions are set to be diagonal Gaussian. We implement them using multi-layer perceptrons (MLP) with the amortization technique and the reparameterization trick (Kingma & Welling, 2013), which take the conditionals as input and output the parameters of the Gaussian.

3 Related Work

Few-shot segmentation According to different interaction strategies between support and query images, existing few-shot segmentation methods can be grouped into prototype-based methods (Shaban et al., 2017; Dong & Xing, 2018; Zhang et al., 2020; 2019b; Wang et al., 2019; Yang et al., 2020; Liu et al., 2020; Tian et al., 2020) and graph-based methods. Dong et al (Dong & Xing, 2018) introduce the first prototype-based FSS method by extending the PrototypicalNet (Snell et al., 2017). SG-One (Zhang et al., 2020) and PANet (Wang et al., 2019) adopt cosine similarity as the interaction method between a single support prototype and the query feature. PFENet (Tian et al., 2020) further propose an effective prior mask and pyramid feature enhancement module to achieve better segmentation performance. Recently, some graph-based methods (Wang et al., 2020; Zhang et al., 2019a; Min et al., 2021; Zhang et al., 2021a) have been proposed to further boost the FSS performance over the prototype-based methods. HSNet (Min et al., 2021) leverage multi-level feature correlation and efficient 4D convolutions to achieve dense comparison between the support and query features. CyCTR (Zhang et al., 2021a) adopt vision transformer with a cycle-consistency attention to aggregate pixel-wise support features into query ones. However, these methods neglect a more realistic setting, where samples of novel classes arrives online.

Online learning Online learning aims to learn from a sequence of data instances one by one and maximizes the correctness for the sequence of predictions (Hoi et al., 2021). Various approaches, such as linear models (Cesa-Bianchi & Lugosi, 2006), non-linear models with kernels (Jin et al., 2010; Kivinen et al., 2004), and deep neural networks (Zhou et al., 2012), have been proposed to tackle the online learning task. Recently, online meta-learning methods (Finn et al., 2019; Babu et al., 2021; Ren et al., 2020) make the model adaptation to the new data faster and more efficiently by leveraging previously seen data. Finn et al. (Finn et al., 2019) propose to achieve fast model adaption to new data with a data buffer storing all task data. Babu et al. (Babu et al., 2021) design a layer-distributed memory network to learn fast adaption. Inspired by the above methods, we extend the traditional few-shot segmentation task to the online setting.

Memory-augmented learning Neural networks with memory exhibit superior capacity in machine learning. Recent work equip the neural network with an external memory module to improve learning capacity Bornschein et al. (2017); Graves et al. (2016); Ramalho & Garnelo (2019); Santoro et al. (2016). For the few-shot scenario, some work with memory network attempt to store the information contained in the support set, focusing on learning the access mechanism shared across tasks Zhen et al. (2020). Memory network exhibits more importance in the online learning setting Santoro et al. (2016); Babu et al. (2021); Ren et al. (2020). Santoro et al. (2016) adopt Neural Turing Machine (NTM) to quickly encode and retrieve new information for the online meta-learning task. Ren et al. Ren et al. (2020) propose a contextual prototypical memory network that can make use of spatiotemporal context from the recent past to tackle the online few-shot classification. In this paper, we propose a prototype memory well-designed for online few-shot semantic segmentation.

Test-time adaptation Test-time adaptation (Sun et al., 2020; Wang et al., 2021; Chen et al., 2022) is proposed to handle domain shifts between training and test data. These algorithms aim to adapt the model to test data by fine-tuning the model parameters with self-supervised loss (Sun et al., 2020; Hu et al., 2021; Zhang et al., 2022) or entropy minimization (Wang et al., 2021; Zhang et al., 2021b; Niu et al., 2022). Recently, Dubey et al. (2021) and Iwasawa & Matsuo (2021) proposed to generate test domain-specific classifiers with batches of test samples. Xiao et al. (2022) further propose to learn to generalize across domains with domain information in single test sample. Different from previous works, we propose variational test-time adaptation on each sample to deal with the sample diversity in online few-shot segmentation, without any fine-tuning on the parameters.

4 Experiments

4.1 Experimental Setup

Datasets We adopt two natural image datasets, i.e., PASCAL and COCO, and one medical dataset ABD-MRI-20, to comprehensively evaluate the proposed method's performance. PASCAL is created from PASCAL VOC 2012 (Everingham et al., 2010) and additional SBD annotations (Hariharan et al., 2011). It contains 20 classes, split into 15 training and 5 testing classes. COCO is a more challenging dataset in which samples from the same classes usually exhibit large appearance and scale variations. It is composed of 60 training classes and 20 testing classes. ABD-MRI-20 (Kavur et al., 2021) is an MRI dataset, which contains 20 3D T2-SPIR MRI scans and each with four organs, i.e., Liver, Spleen, and left and right kidney. We adopt 5 scans with the spleen for evaluation and the remaining 15 scans with other organs for training. T image-mask pairs are randomly sampled from a specific class to construct a data sequence. More details about datasets setup are provided in Appendix B. We will release our code.

Evaluation Metrics We adopt mIoU (mean Intersection over union) as a metric for evaluation on two natural image datasets and dice score on the medical dataset. Given an input sequence with length T, the model makes a random guess on the first image and outputs predicted masks for the remaining images in sequence. We compute the t-shot mIoU (or dice score), i.e., the performance after seeing t image-mask pairs in the sequence. We set the length of the input sequence T as 6 for training and 11 for evaluation, and report the results of 1-shot, 3-shot, 5-shot, 7-shot, and 9-shot. To characterize the learning ability of O-FSS models over sequences, we also present the averaged mIoU from 1-shot to 10-shot results. All numbers are reported with 1000 sequences for natural image datasets and 100 sequences for a medical dataset.

4.2 Baseline Models

To demonstrate the merits of our proposed model, we compared it with several baseline models. These baseline models are built on PFENet (Tian et al., 2020), we extend it to the online version with different category prototype updating strategies. More details can be found in Appendix B.

Online MatchingNet (OMN) MatchingNet (Vinyals et al., 2016) performs nearest neighbour matching among example prototypes for few-shot classification. To make full of category knowledge from previous samples for the segmentation of the current sample, we store the foreground prototypes of previous examples in the prototype memory. In particular, OMN adopts the nearest neighbour matching to select one prototype from the prototype memory, then the most similar prototype serves as category prototype for the segmentation of future samples.

Table 1: Benefit of prototype memory in (%) on PASCAL with ResNet50 averaged three runs. With the prototype memory, our method achieves significant mIoU performance.

Settings	1-shot	3-shot	5-shot	7-shot	9-shot	mean
w/o Prototype memory	57.10 ±2.84	58.37 ±1.33	58.23 ±1.65	57.48 ±0.15	56.67 ±1.62	57.23 ±0.14
w/ **Prototype memory**	**59.41 ±2.50**	**62.82 ±2.31**	**62.91 ±1.29**	**62.91 ±0.16**	**61.73 ±2.16**	**61.83 ±0.10**

Table 2: Variational vs. deterministic classifier in (%) on PASCAL with ResNet50 averaged three runs. Variational classifier is more critical than the deterministic classifier.

Settings	1-shot	3-shot	5-shot	7-shot	9-shot	mean
Deterministic classifier	54.25 ±1.75	58.23 ±2.38	59.41 ±1.17	59.81 ±0.69	58.41 ±1.48	57.82 ±0.04
Variational classifier	**54.84 ±1.37**	**58.96 ±2.87**	**60.03 ±0.78**	**60.30 ±0.39**	**59.11 ±0.96**	**58.53 ±0.18**

Table 3: Importance of test-time adaptation in (%) on PASCAL with ResNet50 averaged three runs. With test-time adaptation, our model achieves clear performance gain.

Settings	1-shot	3-shot	5-shot	7-shot	9-shot	mean
w/o test-time adaptation	54.84 ±1.37	58.96 ±2.87	60.03 ±0.78	60.30 ±0.39	59.11 ±0.96	58.53 ±0.18
w/ test-time adaptation	**59.41 ±2.50**	**62.82 ±2.31**	**62.91 ±1.29**	**62.91 ±0.16**	**61.73 ±2.16**	**61.83 ±0.10**

Online Prototypical Network (OPN) Ren et al. (Ren et al., 2020) extend the Prototypical Network (Snell et al., 2017) to the online setting, where the prototypes are updated sequentially using weighted averaging. For the segmentation task at t step, OPN aggregates the prototypes in the prototype memory into the category prototype by sample averaging. In this way, all previous examples contribute equally to the current segmentation task.

Online Attentive Prototypical Network (OAPN) OAPN assumes that each previous sample contributes differently to the segmentation of the current sample. Specifically, the attention mechanism Vaswani et al. (2017) is adopted to measure the similarity between the prototype of the current sample and prototypes in the prototype memory. Then the category prototype is updated as the weighted sum of prototypes in the memory.

LSTM (Hochreiter & Schmidhuber, 1997) We include temporal modelling methods for comparison as well. Santoro et al. (Santoro et al., 2016) use LSTM with read and write protocols for the online few-shot learning task. Similarly, we adopt a single-layer LSTM to interact with the prototype memory and update the category prototype iteratively.

4.3 Results

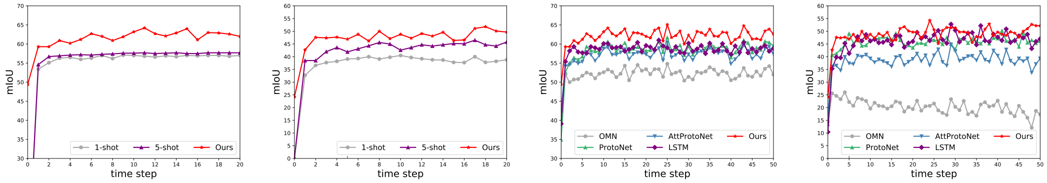
Benefit of prototype memory To show the importance of the prototype memory, we implement a model variant without prototype memory. We directly replace the prototype memory in Eq. 4 with the sample prototype from last time step; that is, we utilize p_{t-1} to generate prior distribution $p(w_t|x_t, p_{t-1})$ and perform the segmentation task of the sample x_t. The experimental results are reported in Table 1. Without the prototype memory, our model has difficulty in aggregating category information from previous samples and adapting to new samples. Thus the performance in all shots is worse than in the memory-based model.

Variational vs. deterministic classifier We compare against the deterministic classifier as our baseline model in which few-shot segmentation training methods obtain the classifier. As shown in Table 2, the proposed variational classifier consistently outperforms the deterministic classifier demonstrating the benefit brought by probabilistic modeling. The variational classifier provides more informative representations of classes, which are able to encompass large intra-class variations and, therefore, improve performance.

Importance of test-time adaptation We investigate the benefit of the test-time adaptation on the PASCAL dataset in Table 3. In this paper, the **memory adaptation** in Eq. 2 is without test-time adaptation, which is only conditioning on the prototype memory, while the **sample adaptation** in

Table 4: Benefit of memory-augmented segmentation in (%) on PASCAL with ResNet50 averaged three runs. Memory-augmented segmentation achieves better performance than with prototype-augmented segmentation.

Settings	1-shot	3-shot	5-shot	7-shot	9-shot	mean
Prototype-augmented	56.17 ±1.32	57.06 ±2.86	59.05 ±0.87	60.78 ±0.68	58.64 ±0.96	59.10 ±0.18
Memory-augmented	**59.41** ±2.50	**62.82** ±2.31	**62.91** ±1.29	**62.91** ±0.16	**61.73** ±2.16	**61.83** ±0.10



(a) PASCAL-21 steps (b) COCO-21 steps (c) PASCAL-51 steps (d) COCO-51 steps

Figure 3: Segmentation performance on long sequences. For sequences with length 21 and 51, the proposed method consistently outperforms traditional and online few-shot segmentation baselines on both PASCAL and COCO datasets.

Eq. 3 is with test-time adaptation, which conditions on both the current sample and the prototype memory. From Table 3, we see that incorporating the variational classifier with test-time adaptation performs consistently better than that without test-time adaptation. This is because, with the test-time adaptation mechanism, our model can learn the capability to adapt to the segmentation of the current sample using sample-specific knowledge from current samples and category information from previous samples.

Benefit of memory-augmented segmentation We demonstrate the benefits of memory-augmented segmentation on the PASCAL dataset. We implement a prototype-augmented variant of our model by replacing the category prototype in Eq. 5 with the sample prototype from the last time step. As shown in Figure 4, our model with memory-augmented segmentation performs consistently better than that with prototype-augmented segmentation. The comparison demonstrates that introducing category knowledge from prototype memory to representation learning is beneficial for better adaptation to the segmentation task of new samples.

Segmentation of long sequences We investigate model performance on long sequences by increasing time steps to 21 and 51, respectively. In Figure 3 (a) and (b), we compare with traditional few-shot segmentation models (more details can be found in Appendix B) trained under 1-shot and 5-shot settings. Interestingly, a simple extension of traditional few-shot segmentation does not cope well with sequential data loading, and tends to converge to over-smoothed, averaged masks of lesser accuracy. In Figure 3 (a) and (b), we increase the time step to 50 and compare it with four online few-shot segmentation baseline models. Our model achieves superior performance than the four baseline variants with time step increases. Our model consistently outperforms traditional few-shot segmentation models. This is because of the variational test-time adaptation mechanism, which dynamically adapts the model to new samples in the sequence.

Comparison with baseline models As shown in Table 5, the proposed method sets consistent state-of-the-art performance on all online few-shot segmentation benchmarks. For instance, our model surpasses the second-best method on PASCAL, i.e., LSTM, by a margin of 3.54% in terms of mean mIoU. This is reasonable since we generate model parameters with sample-specific knowledge from the current sample and category knowledge from previous samples, leading to more adapted models. Furthermore, we provide visualization of online few-shot segmentation results in Figure 4. We can conclude that our model is able to adapt to each new samples and make better mask prediction as time steps increase. More experimental results can be found in Appedix C.

Table 5: Comparison with baseline models on three datasets. PASCAL and COCO adopt mIoU as metric, while ABD-MRI-20 uses dice score, mean value and variance are reported with three runs. Our model is a consistent top-performer on both natural image and medical image datasets.

Dataset	Method	1-shot	3-shot	5-shot	7-shot	9-shot	mean
PASCAL	OMN	51.60 ± 2.11	53.82 ± 4.68	53.62 ± 3.16	53.10 ± 1.17	51.94 ± 3.36	52.62 ± 0.03
	OPN	52.63 ± 3.74	57.96 ± 2.61	59.30 ± 1.89	59.53 ± 0.10	58.60 ± 1.51	57.55 ± 0.05
	OAPN	54.14 ± 1.17	57.30 ± 2.55	58.12 ± 2.29	58.45 ± 0.75	57.13 ± 1.25	56.76 ± 0.02
	LSTM	55.40 ± 2.79	58.96 ± 3.36	59.63 ± 0.74	60.03 ± 0.22	58.50 ± 0.17	58.29 ± 0.10
	Ours	**59.41 ±2.50**	**62.82 ±2.31**	**62.91 ±1.29**	**62.91 ±0.16**	**61.73 ±2.16**	**61.83 ±0.10**
COCO	OMN	23.4 ±4.18	22.42 ±0.68	21.73 ±0.07	23.30 ±3.83	22.48 ±0.03	22.84 ±0.78
	OPN	39.59 ±1.87	42.60 ±1.40	45.22 ±0.36	45.11 ±0.39	47.06 ±3.40	44.22 ±0.20
	OAPN	35.36 ±5.46	36.83 ±3.19	37.99 ±1.87	38.49 ±4.18	40.42 ±3.20	38.19 ±0.66
	LSTM	35.52 ±1.89	41.45 ±3.32	44.65 ±1.36	45.67 ±0.55	47.71 ±5.13	43.84 ±0.58
	Ours	**43.08 ±1.61**	**45.96 ±1.18**	**49.17 ±0.5**	**48.30 ±0.14**	**49.93 ±3.24**	**47.79 ±0.29**
ABD-MRI-20	OMN	32.28 ±2.24	30.16 ±1.16	24.43 ±1.15	30.74 ±0.29	26.76 ±1.18	28.36 ±0.36
	OPN	35.40 ±1.27	30.95 ±0.23	36.86 ±0.36	35.89 ±0.01	33.15 ±0.29	34.29 ±0.08
	OAPN	37.33 ±1.81	32.17 ±0.20	38.39 ±0.28	38.20 ±1.03	34.82 ±0.74	35.84 ±0.23
	LSTM	34.66 ±1.40	29.08 ±0.10	35.82 ±0.23	33.97 ±1.55	31.72 ±0.80	32.74 ±0.18
	Ours	**39.57 ±0.58**	**34.48 ±0.61**	**41.26 ±0.09**	**39.73 ±0.71**	**36.53 ±2.05**	**38.32 ±0.28**

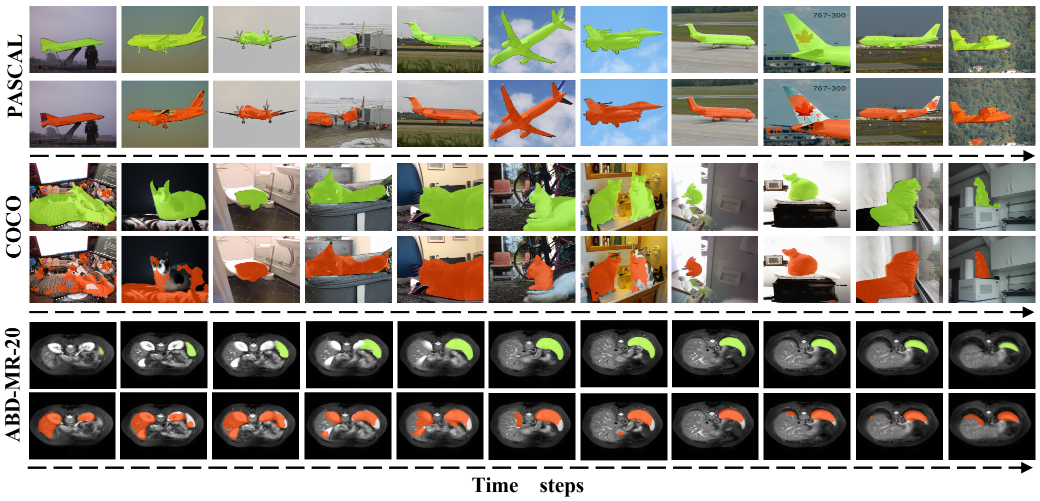


Figure 4: Visualization of online few-shot segmentation performance on PASCAL (top), COCO (middle), and ABD-MRI-20 (bottom). Ground-truths are masked in green, while predictions are masked in red. As time steps increase, our model is able to make better mask prediction.

5 Conclusion

In this paper, we investigate online few-shot segmentation, which aims to make pixel-wise prediction for samples from novel classes sequentially. To cope with large sample diversity in the sequence, we propose a memory-augmented variational adaptation mechanism, which adapts model to each new sample. We first propose a prototype memory to retain category knowledge from previous samples, then formulate the adaptation to the sample as a variational Bayesian inference problem. Conditioned on the current sample and an external memory, our method is able to generate sample-specific classifiers for the sample at each time step. Furthermore, we propose memory-augmented segmentation to learn sample-specific representation for each sample. By doing so, our method is updated sequentially and achieves fast adaptation to each sample segmentation task with the number of samples increases over time. Ablation studies and further experiments on both natural image and medical datasets show that our method attains superior online few-shot segmentation performance.

References

Sudarshan Babu, Pedro Savarese, and Michael Maire. Online meta-learning via learning with layer-distributed memory. *Advances in Neural Information Processing Systems*, 34:14795–14808, 2021.

Jörg Bornschein, Andriy Mnih, Daniel Zoran, and Danilo Jimenez Rezende. Variational memory addressing in generative models. *Advances in Neural Information Processing Systems*, 30, 2017.

Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. Contrastive test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 295–305, 2022.

Nanqing Dong and Eric P Xing. Few-shot semantic segmentation with prototype learning. In *BMVC*, 2018.

Abhimanyu Dubey, Vignesh Ramanathan, Alex Pentland, and Dhruv Mahajan. Adaptive methods for real-world domain generalization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 14340–14349, 2021.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2): 303–338, 2010.

Chelsea Finn, Aravind Rajeswaran, Sham Kakade, and Sergey Levine. Online meta-learning. In *International Conference on Machine Learning*, pp. 1920–1930. PMLR, 2019.

Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, et al. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538 (7626):471–476, 2016.

Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *2011 International Conference on Computer Vision*, pp. 991–998. IEEE, 2011.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Steven CH Hoi, Doyen Sahoo, Jing Lu, and Peilin Zhao. Online learning: A comprehensive survey. *Neurocomputing*, 459:249–289, 2021.

Minhao Hu, Tao Song, Yujun Gu, Xiangde Luo, Jieneng Chen, Yinan Chen, Ya Zhang, and Shaoting Zhang. Fully test-time adaptation for image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 251–260. Springer, 2021.

Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Rong Jin, Steven CH Hoi, and Tianbao Yang. Online multiple kernel learning: Algorithms and mistake bounds. In *International conference on algorithmic learning theory*, pp. 390–404. Springer, 2010.

A Emre Kavur, N Sinem Gezer, Mustafa Barış, Sinem Aslan, Pierre-Henri Conze, Vladimir Groza, Duc Duy Pham, Soumick Chatterjee, Philipp Ernst, Savaş Özkan, et al. Chaos challenge-combined (ct-mr) healthy abdominal organ segmentation. *Medical Image Analysis*, 69:101950, 2021.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Jyrki Kivinen, Alexander J Smola, and Robert C Williamson. Online learning with kernels. *IEEE transactions on signal processing*, 52(8):2165–2176, 2004.

Jie Liu, Yanqi Bao, Guo-Sen Xie, Huan Xiong, Jan-Jakob Sonke, and Efstratios Gavves. Dynamic prototype convolution network for few-shot semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11553–11562, 2022.

Yongfei Liu, Xiangyi Zhang, Songyang Zhang, and Xuming He. Part-aware prototype network for few-shot semantic segmentation. In *European Conference on Computer Vision*, pp. 142–158. Springer, 2020.

Xiangde Luo, Guotai Wang, Tao Song, Jingyang Zhang, Michael Aertsen, Jan Deprest, Sebastien Ourselin, Tom Vercauteren, and Shaoting Zhang. Mideepseg: Minimally interactive segmentation of unseen objects from medical images using deep learning. *Medical Image Analysis*, 72:102102, 2021.

Juhong Min, Dahyun Kang, and Minsu Cho. Hypercorrelation squeeze for few-shot segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6941–6952, 2021.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. *arXiv preprint arXiv:2204.02610*, 2022.

Tiago Ramalho and Marta Garnelo. Adaptive posterior learning: few-shot learning with a surprise-based memory module. *arXiv preprint arXiv:1902.02527*, 2019.

Mengye Ren, Michael L Iuzzolino, Michael C Mozer, and Richard S Zemel. Wandering within a world: Online contextualized few-shot learning. *arXiv preprint arXiv:2007.04546*, 2020.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pp. 1842–1850. PMLR, 2016.

Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. *arXiv preprint arXiv:1709.03410*, 2017.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning*, pp. 9229–9248. PMLR, 2020.

Zhuotao Tian, Hengshuang Zhao, Michelle Shu, Zhicheng Yang, Ruiyu Li, and Jiaya Jia. Prior guided feature enrichment network for few-shot segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021.

Haochen Wang, Xudong Zhang, Yutao Hu, Yandan Yang, Xianbin Cao, and Xiantong Zhen. Few-shot semantic segmentation with democratic attention networks. In *European Conference on Computer Vision*, pp. 730–746. Springer, 2020.

Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. Panet: Few-shot image semantic segmentation with prototype alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9197–9206, 2019.

Zehao Xiao, Xiantong Zhen, Ling Shao, and Cees G M Snoek. Learning to generalize across domains on single test samples. In *International Conference on Learning Representations*, 2022.

Boyu Yang, Chang Liu, Bohao Li, Jianbin Jiao, and Qixiang Ye. Prototype mixture models for few-shot semantic segmentation. In *European Conference on Computer Vision*, pp. 763–778. Springer, 2020.

Chi Zhang, Guosheng Lin, Fayao Liu, Jiushuang Guo, Qingyao Wu, and Rui Yao. Pyramid graph networks with connection attentions for region-based one-shot semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9587–9595, 2019a.

Chi Zhang, Guosheng Lin, Fayao Liu, Rui Yao, and Chunhua Shen. Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5217–5226, 2019b.

Gengwei Zhang, Guoliang Kang, Yi Yang, and Yunchao Wei. Few-shot segmentation via cycle-consistent transformer. *Advances in Neural Information Processing Systems*, 34, 2021a.

Marvin Zhang, Sergey Levine, and Chelsea Finn. Memo: Test time robustness via adaptation and augmentation. *arXiv preprint arXiv:2110.09506*, 2021b.

Xiaolin Zhang, Yunchao Wei, Yi Yang, and Thomas S Huang. Sg-one: Similarity guidance network for one-shot semantic segmentation. *IEEE Transactions on Cybernetics*, 50(9):3855–3865, 2020.

Yizhe Zhang, Shubhankar Borse, Hong Cai, and Fatih Porikli. Auxadapt: Stable and efficient test-time adaptation for temporally consistent video semantic segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2339–2348, 2022.

Xiantong Zhen, Yingjun Du, Huan Xiong, Qiang Qiu, Cees Snoek, and Ling Shao. Learning to learn variational semantic memory. *Advances in Neural Information Processing Systems*, 33: 9122–9134, 2020.

Guanyu Zhou, Kihyuk Sohn, and Honglak Lee. Online incremental feature learning with denoising autoencoders. In *Artificial intelligence and statistics*, pp. 1453–1461. PMLR, 2012.

A Derivations of ELBO

For the sample x_t at time step t, we begin with maximizing log-likelihood of the conditional distribution $\log p(y_t|x_t, \mathcal{M}_t)$ to derive the ELBO. By applying Jensen's inequality, we have the following steps as

$$\begin{aligned}
&\log p(y_t|x_t, \mathcal{M}_t) \\
&= \log \int p(y_t|x_t, w_t)p(w_t|x_t, \mathcal{M}_t)dw_t \\
&= \log \int p(y_t|x_t, w_t)\frac{p(w_t|x_t, \mathcal{M}_t)}{q(w_t|x_t, y_t, \mathcal{M}_t)}q(w_t|x_t, y_t, \mathcal{M}_t)dw_t \\
&\geq \int \log[\frac{p(y_t|x_t, w_t)p(w_t|x_t, \mathcal{M}_t)}{q(w_t|x_t, y_t, \mathcal{M}_t)}]q(w_t|x_t, y_t, \mathcal{M}_t)dw_t \\
&= \mathbb{E}_{q(w_t)}[p(y_t|x_t, w_t)] - \mathbb{D}_{KL}[q(w_t|x_t, y_t, \mathcal{M}_t)||p(w_t|x_t, \mathcal{M}_t)],
\end{aligned} \tag{9}$$

which is consistent with Eq. 4.

B Implementation

B.1 Datasets details

Pascal-5^i and COCO-20^i are two widely-used benchmarks in traditional few-shot segmentation (FSS). Cross validation is adopted in FSS to test model performance on different novel classes, we provide the class split of different folds in Table 6 and Table 7, respectively. In online few-shot segmentation (OFSS), we adopt two nature image dataest (PASCAL and COCO) and one medical image dataset ABD-MR-20 to verify the effectiveness of online few-shot segmentation models. For PASCAL and COCO, we implement most experiments on the fold-0 of Pascal-5^i and COCO-20^i, i.e., classes in fold-0 serve as testing classes, while remaining classes are training classes. We also provide results on different folds in Table 11, and more detailed results can be found in Table. ABD-MRI-20 is a MRI dataset from ISBI 2019 Combined Healthy Abdominal Organ Segmentation Challenge (Kavur et al., 2021). We choose spleen as the testing class, Liver, left and right kidney as training classes. Furthermore, we adopt 5 scans with spleen for evaluation and remaining 15 scans with other organs for training.

Table 6: Testing classes split for each fold in PASCAL-5^i dataset.

Fold	Testing (novel) classes
Fold-0	Aeroplane, Bicycle, Bird, Boat, Bottle
Fold-1	Bus, Car, Cat, Chair, Cow
Fold-2	Diningtable, Dog, Horse, Motorbike, Person
Fold-3	Potted plant, Sheep, Sofa, Train, Tvmonitor

Table 7: Testing classes split for each fold in COCO-20^i dataset.

Fold	Testing (novel) classes
Fold-0	Person, Airplane, Boat, Parking meter, Dog, Elephant, Backpack,Suitcase, Sports Ball, Skateboard, Wine glass, Spoon, Sandwich, Hot dog, Chair, Dining table, Mouse, Microwave, Scissorse
Fold-1	Bicycle, Bus, Traffic light, Bench, Horse, Bear, Umbrella, Frisbee, Kite, Surfboard , Cup, Bowl, Orange, Pizza, Couch, Toilet, Remote, Oven, Book, Teddy bear
Fold-2	Car, Train, Fire hydrant, Bird, Sheep, Zebra, Handbag, Skis, Baseball bat, Tennis racket, Fork, Banana, Broccoli, Donut, Potted plant, Tv, Keyboard, Sink, Toaster, Clock, Hair drier
Fold-3	Motorcycle, Truck, Stop sign, Cat, Cow, Giraffe, Tie, Snowboard, Baseball glove, Bottle, Knife, Apple, Carrot, Cake, Bed, Laptop, Cell phone, Refrigerator, Vase, Toothbrush

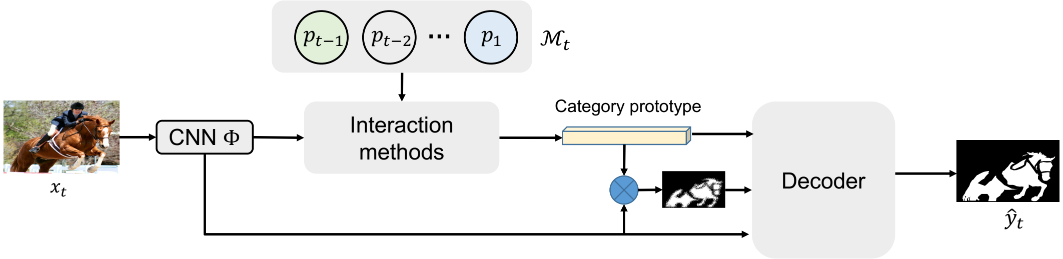


Figure 5: **Architecture of baseline models**. We compare our model with four baseline models, which adopt different interaction methods between the prototype memory $\mathcal{M}_t$ and CNN features of the current sample to generate the category prototype.

B.2 Implementation details

Task setup Online few-shot segmentation takes sequential samples as input and outputs mask prediction for each sample in the sequence. All samples in a specific sequence contain the same class object. Denoting the length of the input sequence as T, we set $T = 6$ and $T = 11$ at the training and testing stage, respectively. For natural image datasets, we randomly sample thousands of sequences from training classes to train our model at the training stage. At the testing stage, we randomly sample 1000 sequences from novel classes to evaluate model performance. The input resolutions of the model is set as 473×473. For the medical dataset, we focus on the segmentation of 2D slices. At the training stage, we first select one 3D MRI scan, then randomly sample T 2D slices that contain the target organ as one sequence. At the testing stage, we set the testing number of sequences as 100, and the input resolution is 200×200.

Training details We train all baseline models and the proposed model with learning rate 0.0025 for 100 and 50 epochs on PASCAL and COCO, respectively. For experiments on ABD-MRI-20, we set the learning rate and training epochs as 0.0025 and 100, respectively. We adopt ResNet50(He et al., 2016) pretrained on ImageNet (Russakovsky et al., 2015) as backbone network to extract features. The backbone is frozen for experiments on PASCAL and COCO to avoid the model outfitting to training classes. For experiments on ABD-MRI-20, we fine-tune the backbone network to learn robust feature representation for medical segmentation.

B.3 Baseline models

In our experiments, we compare our model with four baseline models, i.e., Online matching net (OMN), Online Prototyical Network (OPN), Online Attentive Prototypical Network (OAPN), and LSTM. These four baseline models share the same architecture as shown in Figure 5. The main difference between these baseline models is the interaction methods between the prototype memory $\mathcal{M}_t$ and CNN features of the current sample to generate the category prototype. OMN adopts the nearest neighbour matching between the prototype memory and the feature of the current sample, then the most similar prototypes serves as the category prototype for the segmentation of future samples. OPN obtains the category prototype by averaging sample prototypes in the prototype memory, while OAPN adopts the attention mechanism to generate weights for each prototype in the prototype memory and the category prototype is updated as the weighted sum of prototypes in the memory. For LSTM, we adopt a single-layer LSTM to interact with the prototype memory to update the category prototype.

In Figure 3 (a) and (b), we compare our model with traditional few-shot segmentation models trained under 1-shot and 5-shot settings. Replacing the prototype memory in Figure 5 with the support set, we can obtain our traditional few-shot segmentation models. When the number of samples increases over time, we directly average support foreground prototypes to get the category prototype. For instance, at time step $t = 5$, we first obtain foreground prototypes of previous four samples, then we average four prototypes to get the category prototype, which is finally used to preform the segmentation of the fifth sample.

Table 8: **Per step results on PASCAL**. We report the results from 0-shot to 10-shot and the mean of 1-shot to 10-shot. Our method achieves consistent best performance. mIoU is adopted as metric.

Method	0-shot	1-shot	2-shot	3-shot	4-shot	5-shot	6-shot	7-shot	8-shot	9-shot	10-shot	mean
FSS-1shot	0	53.62	53.27	53.93	53.97	53.44	53.27	53.53	53.47	53.30	53.43	53.51
FSS-5shot	0	53.47	56.40	56.87	57.27	57.17	57.27	57.43	57.33	57.57	57.33	56.91
OMN	42.17	51.60	52.27	53.81	52.47	53.62	52.97	53.10	52.84	51.92	51.71	52.62
OPN	35.47	52.63	55.87	57.97	56.47	59.30	58.62	59.53	58.32	58.64	58.23	57.55
OAPN	43.29	54.15	55.48	57.30	55.90	58.12	57.37	58.45	57.10	57.13	56.63	56.76
LSTM	39.70	55.40	57.37	58.97	57.37	59.63	59.20	60.03	58.37	58/50	58.03	58.29
Ours	49.39	59.41	60.21	62.82	61.42	62.91	62.48	62.91	62.40	61.74	62.06	61.83

Table 9: **Per step results on COCO**.We report the results from 0-shot to 10-shot and the mean of 1-shot to 10-shot. Our method achieves consistent best performance. mIoU is adopted as metric.

Method	0-shot	1-shot	2-shot	3-shot	4-shot	5-shot	6-shot	7-shot	8-shot	9-shot	10-shot	mean
FSS-1shot	0	33.38	37.64	38.42	37.72	38.06	38.86	39.78	39.14	38.63	39.64	38.13
FSS-5shot	0	38.76	41.98	43.09	44.08	43.90	44.67	45.26	45.21	44.08	45.34	43.64
OMN	14.09	23.40	24.34	22.40	23.04	21.73	22.27	23.30	23.01	22.48	22.46	22.84
OPN	11.02	39.59	44.37	42.60	42.53	45.22	44.56	45.11	46.01	47.06	45.13	44.22
OAPN	16.13	35.36	38.70	36.83	38.24	37.99	37.45	38.49	39.54	40.42	38.87	38.19
LSTM	0.09	35.52	41.20	41.45	44.10	44.64	45.00	45.67	47.14	47.71	46.04	43.84
Ours	25.17	43.08	47.57	45.96	46.71	49.17	48.46	48.30	49.90	49.93	48.82	47.79

Table 10: **Per step results on ABD-MRI-20**.We report the results from 0-shot to 10-shot and the mean of 1-shot to 10-shot. Our method achieves consistent best performance. Dice score is adopted as metric.

Method	0-shot	1-shot	2-shot	3-shot	4-shot	5-shot	6-shot	7-shot	8-shot	9-shot	10-shot	mean
OMN	9.33	32.28	30.16	30.46	24.59	24.42	29.12	30.74	25.50	26.76	29.54	28.36
OPN	13.65	35.40	39.72	30.95	34.73	36.86	32.88	35.89	30.78	33.15	32.57	34.29
OAPN	15.20	37.33	42.06	32.17	36.07	38.39	33.83	38.20	31.91	34.82	33.58	35.84
LSTM	12.29	34.66	37.80	29.08	32.23	35.82	30.92	33.97	30.11	31.72	31.07	32.74
Ours	18.42	39.57	44.94	34.48	38.90	41.26	36.53	39.72	34.87	36.53	36.38	38.32

C More results

C.1 Per step results

We report per step results of our model and baseline models in Table 8, Table 9, and Table 10 for PASCAL, COCO, and ABD-MRI-20, respectively. As shown in above Tables, our models achieves considerably better performance than baseline models in all three dataset. Our model achieves substantial performance improvement with time step increases, even though experiences some fluctuation. This attributes to the capacity of our model in generating sample-specific weights for each sample in the sequence. Interestingly, our model also learns to distinguish salient objects from complex backgrounds. For zero-shot segmentation, in which online few-shot segmentation models give random guess on the first image of a specific novel class, our model also achieves best performance.

C.2 Cross validation on different unseen classes

To investigate the effectiveness of our model on different novel classes, we implement cross validation on unseen classes and report results in Table 11. We compare our method with naive classifier implemented with a 1×1 convolutional layer, i.e., test-time adaptation vs. naive classifier. As shown in Table 11, our method achieves the best performance across different folds on both PASCAL and COCO datasets. We can conclude that our model shows superior performance for online few-shot segmentation and is robust to different novel classes.

Table 11: Cross validation on different unseen classes. For each fold, testing samples come from different unseen classes. Our method consistently outperforms baseline method on different folds of PASCAL and COCO datasets.

Settings	PASCAL				COCO			
	Fold-0	Fold-1	Fold-2	Fold-3	Fold-0	Fold-1	Fold-2	Fold-3
Naive classifier	57.82 ±0.04	66.15 ±0.26	52.35 ±0.12	49.66 ±0.26	41.74 ±0.36	37.23 ±0.24	16.43 ±0.45	25.20 ±0.37
Variational Test-time adaptation	**61.83** ±**0.10**	**68.87** ±**0.31**	**53.17** ±**0.05**	**51.46** ±**0.32**	**47.79** ±**0.29**	**41.14** ±**0.38**	**18.63** ±**0.11**	**27.60** ±**0.42**

C.3 VISUALIZATION

We provide more visualization of the segmentation process of our model in dealing with a sequence of samples. Examples are shown in Figure 6 and Figure 7, respectively. We can see from the visualization that our model can effectively tacking the sample diversity problem with providing sample-specific weights for each sample. With time step increases, our model makes more and more accurate predictions on coming samples.

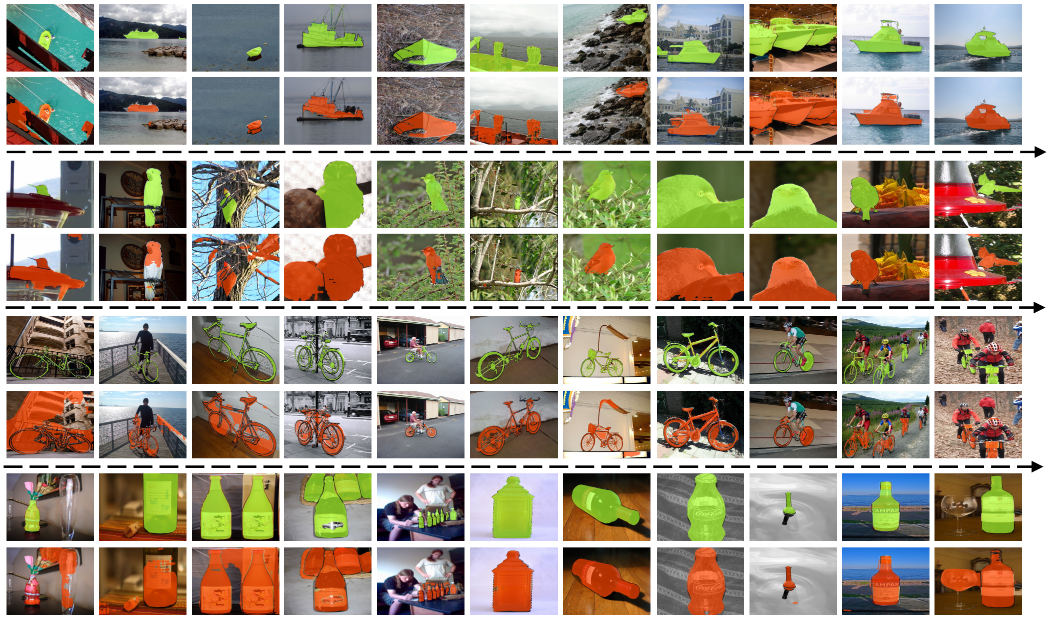


Figure 6: **Visualization of online few-shot segmentation performance on PASCAL.** Ground-truths are masked in green and predictions are masked in red. The length of sequence is set as $T = 11$, and 0-shot to 10-shot results are reported. The input sequence exhibits large sample diversity, our model shows superior capacity in tacking this problem.

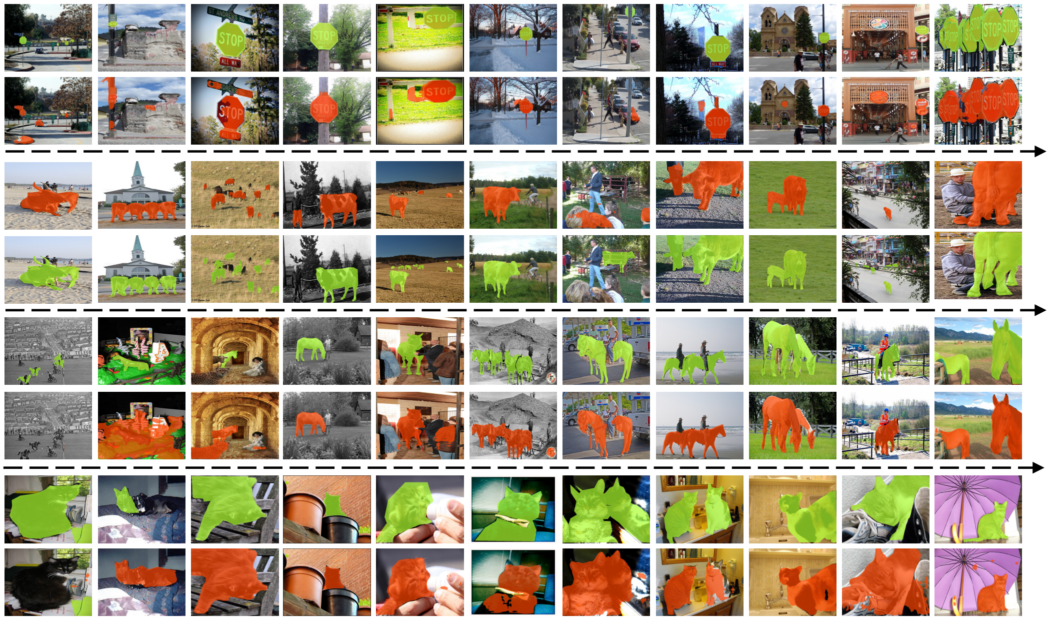


Figure 7: **Visualization of online few-shot segmentation performance on COCO.** Ground-truths are masked in green and predictions are masked in red. The length of sequence is set as $T = 11$, and 0-shot to 10-shot results are reported. The input sequence exhibits large sample diversity, our model shows superior capacity in tacking this problem.